(12) UK Patent Application (19) GB (11) 2 156 115 A

(43) Application published 2 Oct 1985

(21) Application No 8506269

(22) Date of filing 11 March 1985

(30) Priority data
(31) 8406396 (32) 12 Mar 1984 (33) GB

(51) INT CL⁴
G06F 7/04

(52) Domestic classification
G4A CM

(56) Documents cited
GB 1108036 GB 0970725 GB 0969166

(58) Field of search
G4A

(71) Applicant
The Secretary of State for Defence (United Kingdom), Whitehall, London SW1A 2HB

(72) Inventor
Richard Anthony Evans

(74) Agent and/or Address for Service
Dr A W S Williams,
Procurement Executive, Ministry of Defence, Patents 1A4, Room 2014, Empress State Building, Lillie Road, London SW6 1TR

(54) Regular expression recognition apparatus

(57) Regular expression recognition apparatus includes token cells 92, 94, 96, 98, 100, union operation cells 91, 93, 95, 97, 99, 101, 102 and star operation cells (150, 151 and 170, Figs. 4, 5). Each cell contains a result input delay e.g. 114 and means for comparing input and stored characters. An AND gate 115 provides a true output if input and stored characters match and the input result on the previous cycle was true. The token, star and union operation cells are arranged so that regular expressions may be mapped in electronic circuitry on a one-to-one basis in the same sequences in which they are written algebraically. Subsidiary connectios or additional design exercises are not required.

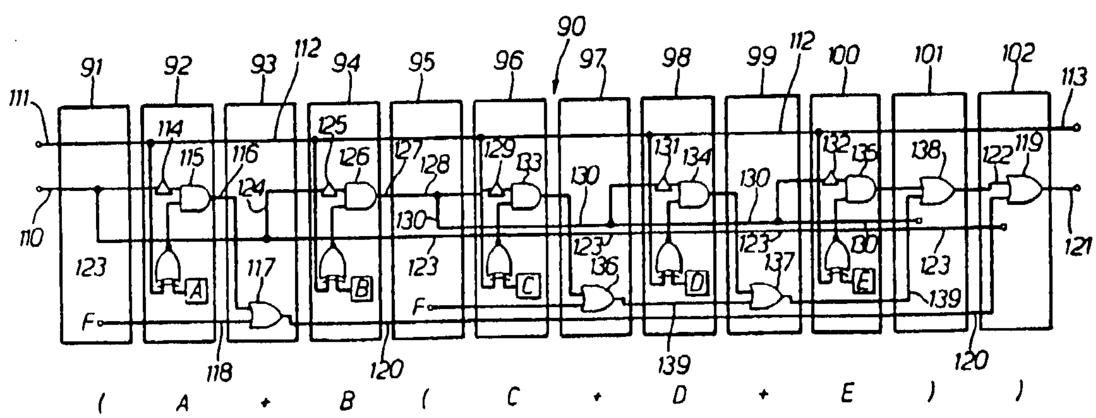


Fig.3.

GB 2 156 115 A

Reprinted front page

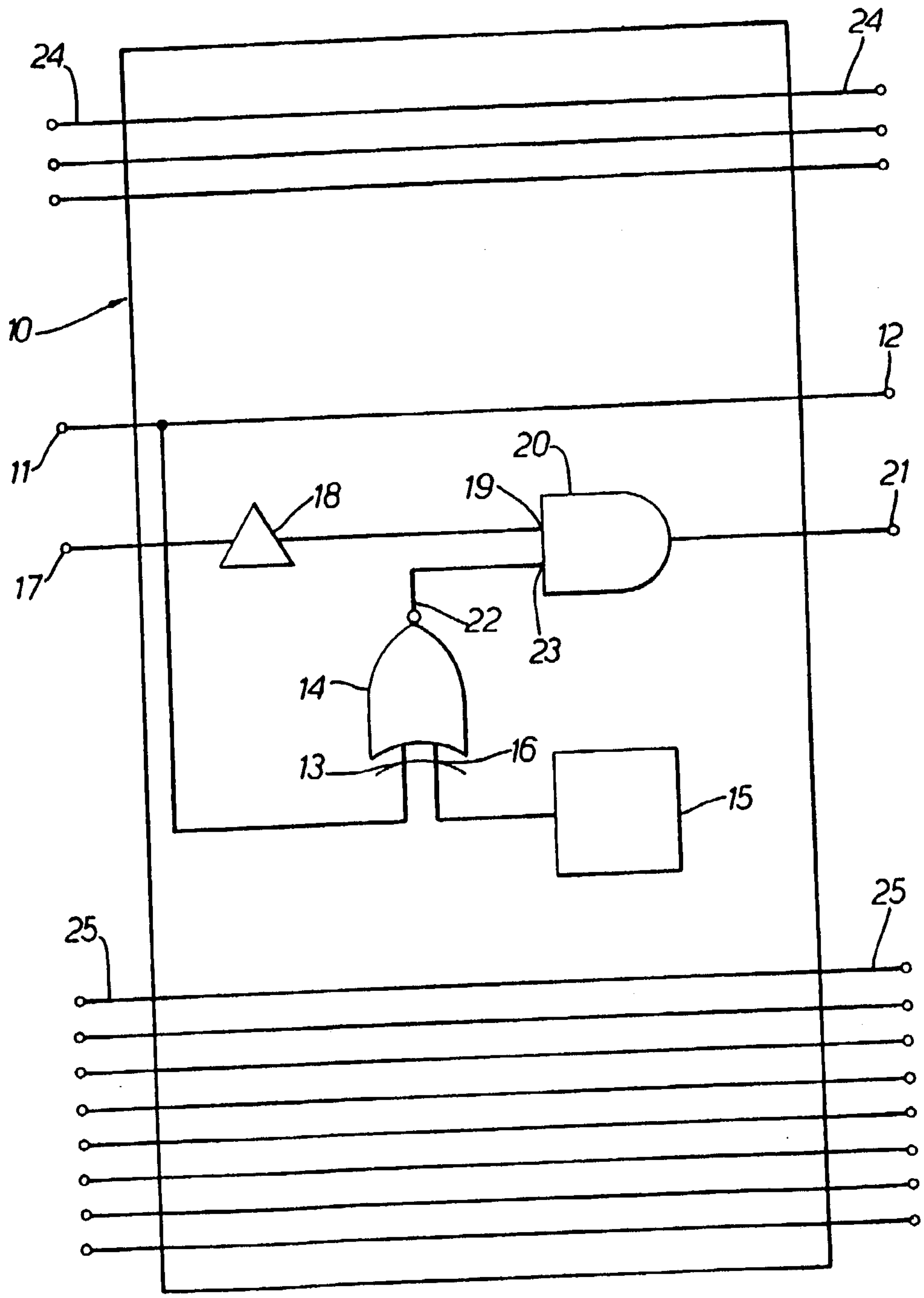


Fig.1.

2156115

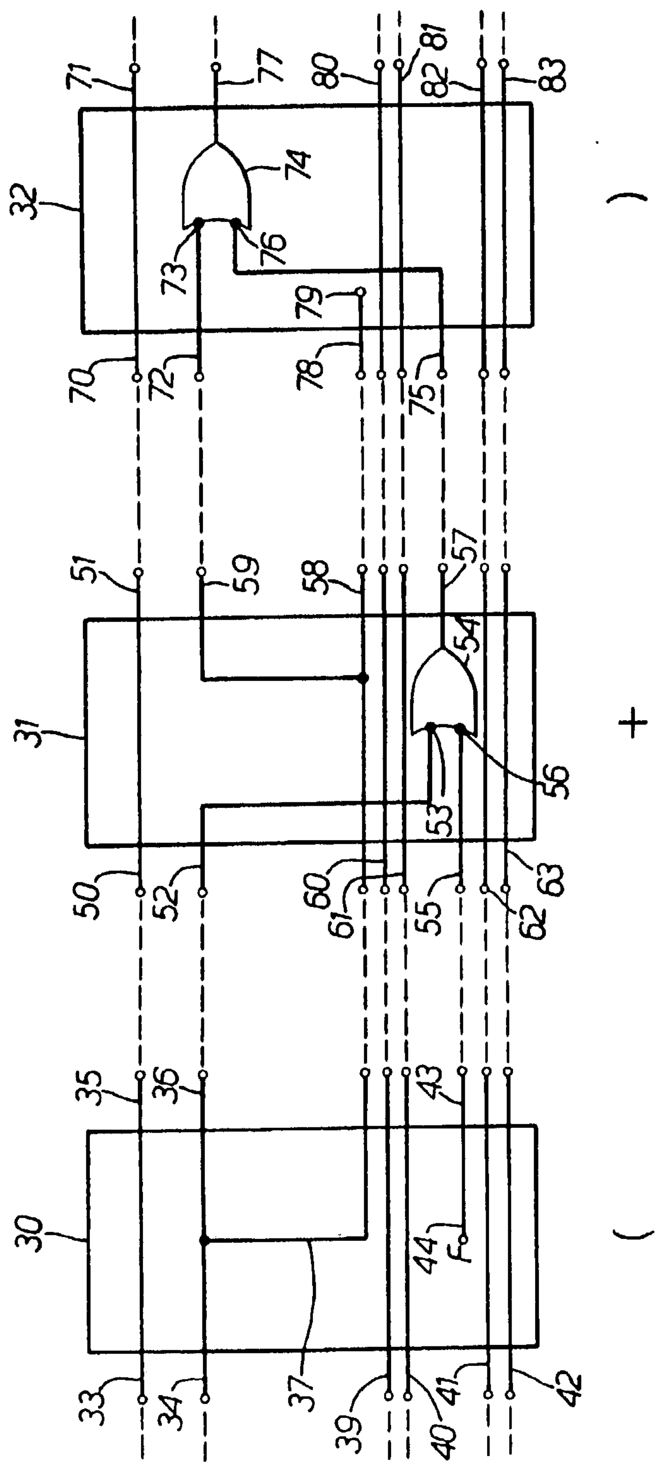
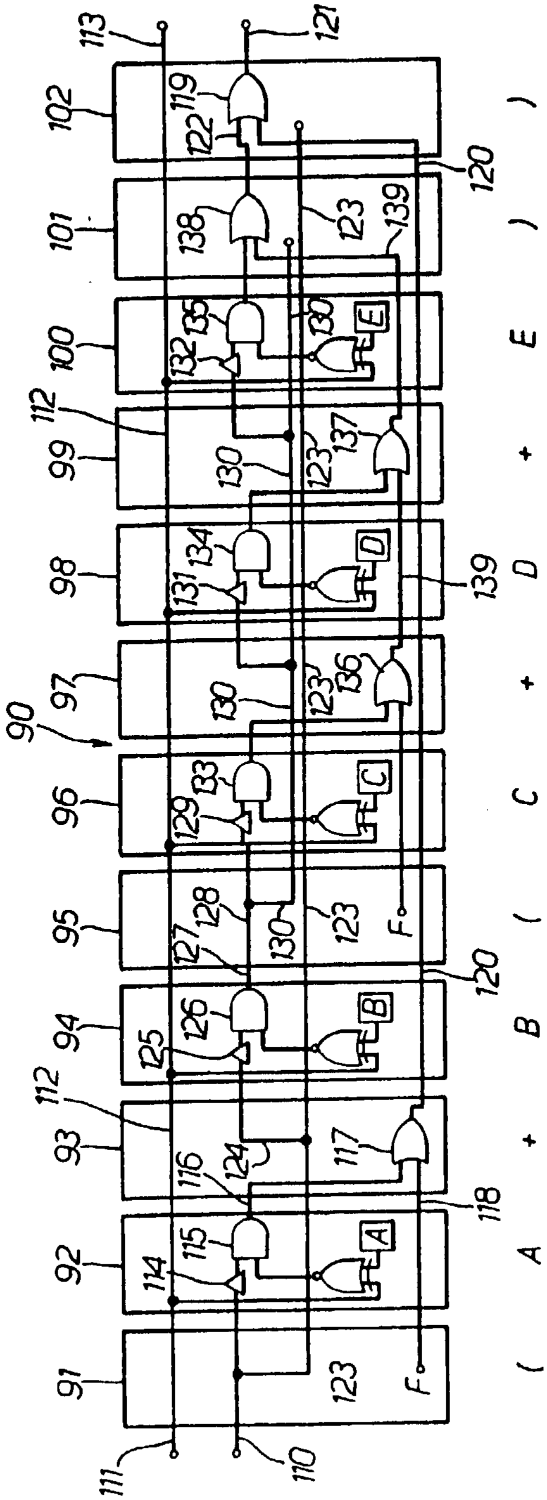


Fig. 2.



90
91
92
93
94
95
96
97
98
99
100
101
102
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
A
B
C
D
E
F
(A + B (C + D + E))

Fig.3.

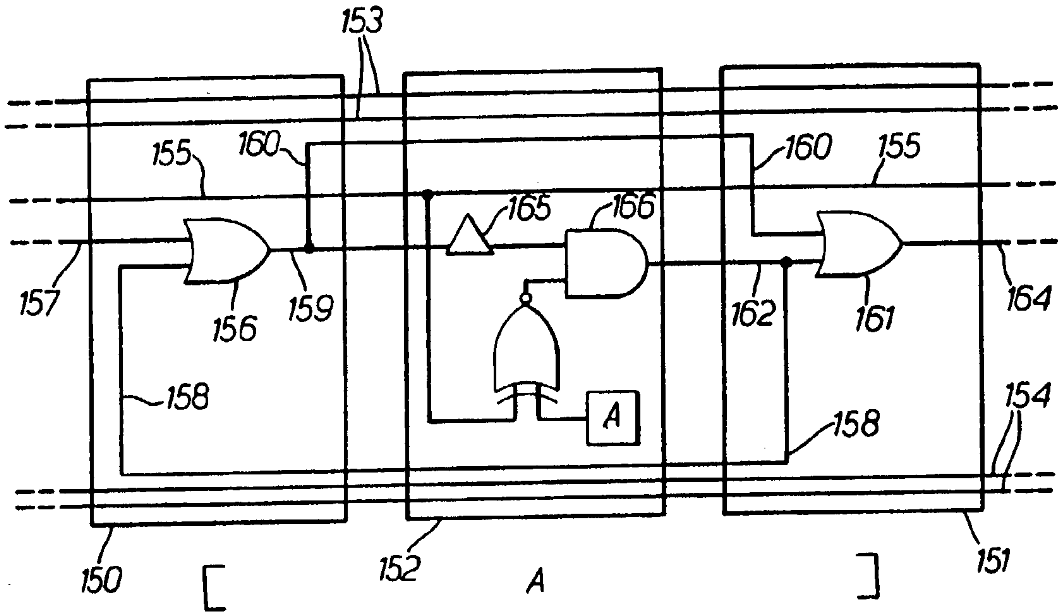

153
155
160
160
155
165
166
157
156
159
162
161
164
158
A
154
158
150
152
151
[
A
]

Fig.4.

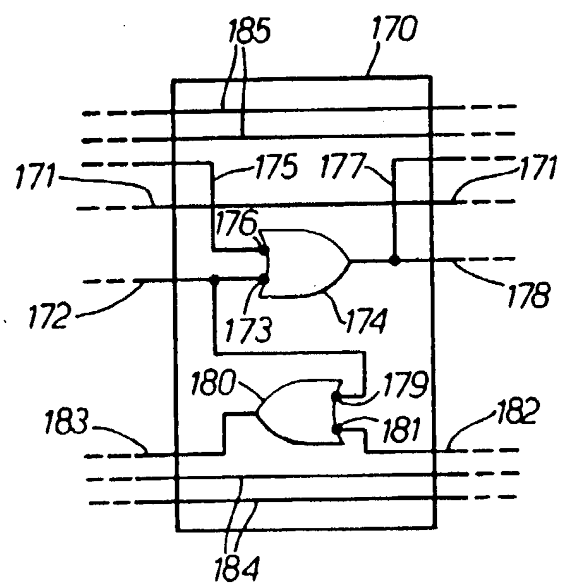
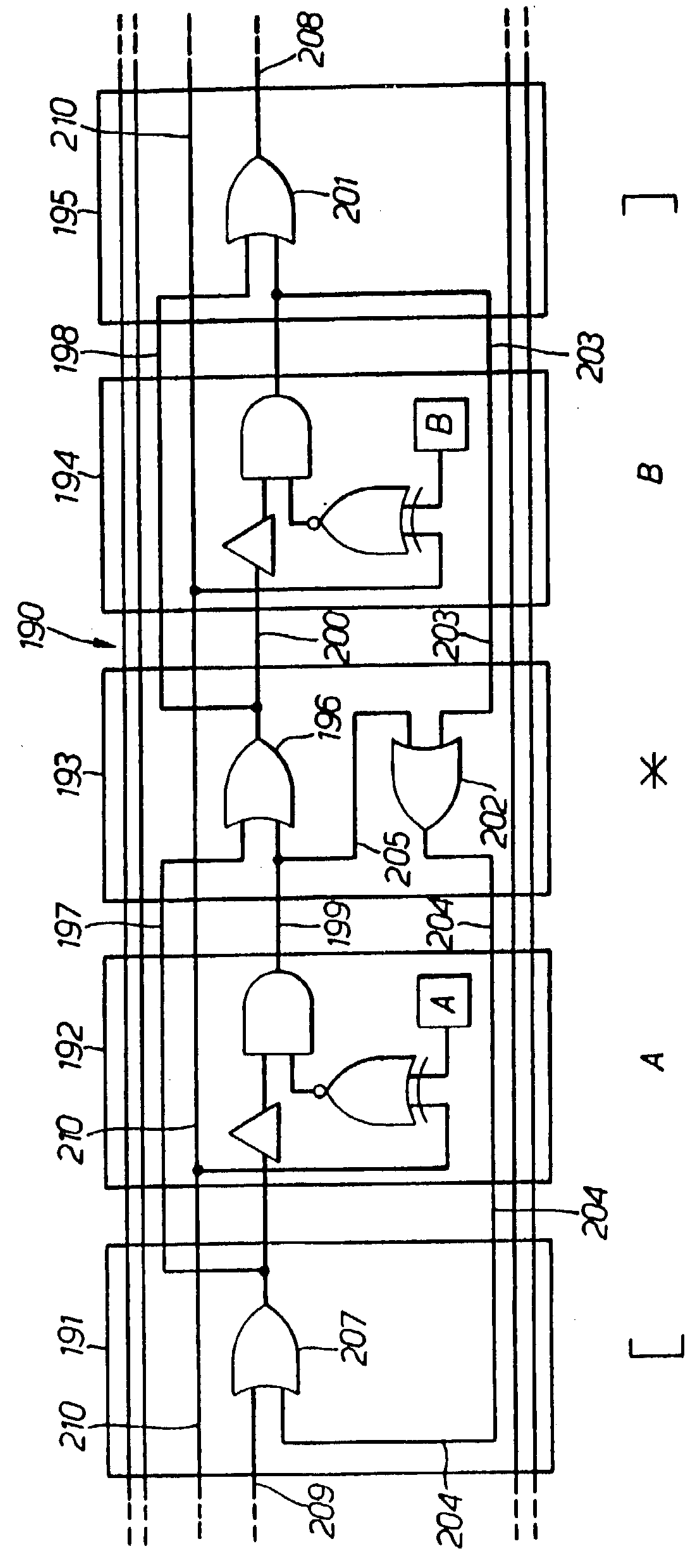

185
170
171
175
177
171
176
172
173
174
178
180
179
183
181
182
184

Fig.5.



208
210
195
201
198
203
194
B
190
200
203
193
196
205
202
197
199
204
192
A
210
204
191
207
210
209
204

Fig. 6.

SPECIFICATION

Regular expression recognition apparatus

This invention relates to regular expression recognition apparatus.

A regular expression is a mathematical concept which is defined with reference to the set of rules it obeys. It is assumed to involve so-called tokens labelled A, B, C etc. A token is a representation of a signal value to be recognised. In digital electronics, a token is a signal in bit-serial or bit-parallel form. A regular expression is defined as involving an arbitrary number of tokens together with an arbitrary number of up to three forms of operator. The simplest form of operation, concatenation of an arbitrary number of tokens to form a string or series, is represented simply by ABC... The union operator provides an OR or + operation in Boolean algebra, ie (A + B) represents either A or B. The star operator, represented by an asterisk suffix to a token, denotes an arbitrary number (including zero) of occurrences of the relevant token. Accordingly A^* denotes n successive occurrences of A, where n is any positive integer including zero.

As an example, the regular expression $(A + B)BCA^* (B + C)^*$ denotes either A or B followed by B and C, then an arbitrary number of A, and finally an arbitrary number of tokens each of which may either be B or C.

General regular expressions are defined recursively as follows:

(a) A string consisting of a single token is a regular expression;

(b) If $X_1, X_2, \ldots X_n$ are regular expressions, so also are:

(1) X_1^*,

(2) $X_1X_2 \ldots X_n$, and

(3) $(X_1 + X_2 + \ldots + X_n)$;

(c) No expression is a regular expression unless it can be formed from some combination of steps (a) and (b).

There are many potential applications for regular expression recognition apparatus. Such expressions have been used to describe hand-written characters, so an appropriate set of recognition devices might be employed in the process of enabling a computer to read manuscript. Telecommunication protocols require recognition of code words, which may be represented as regular expressions. Controlling and sequencing operations might be implemented by employing particular regular expressions to initiate corresponding operations. Recognition of computer instructions might be carried out by regular expression recognition apparatus, an example being a microprocessor recognising microcode. Other examples include database searching for keywords, and editing of computer files by searching for character strings.

In VLSI '83 (Elsevier Science Publishers BV, North-Holland), Curry and Mukhopadhyay observe that the traditional "Von Neumann" digital computer based on a central processing unit is not well suited to regular expression recognition. These authors also describe the design of an integrated circuit suitable for matching simple strings of characters, ie concatenations. The circuit is however comparatively limited in scope, since it is not adapted to deal with all forms of regular expression. It is capable of dealing with single token repetition for example, but not repetition (star operator) of unions or concatenations.

In the Proceedings of VLSI '81 (Academic Press), Foster and Kung describe the use of programmable building blocks or VLSI circuits for regular expression recognition. Each block has a respective function of recognising an aspect of a regular expression. However, the approach described has a number of limitations. In particular, it is not a simple procedure to map a regular expression into a corresponding arrangement of building blocks. The order of building blocks is in general different to that of the terms in an equivalent general expression. Moreover, the building blocks require a separate interconnection scheme. Both the order of blocks and their interconnection scheme require a detailed design analysis for their determination.

Regular expression recognition apparatus necessarily includes one or more token cells of known kind for comparing a token or character signal input with a stored signal value. The token cell indicates a match occurrence when achieved, provided that any earlier tokens in the expression have been matched. A typical token cell might include a memory to store a signal value for matching with an input signal, a signal comparator and an AND gate for indicating match result occurrence if and only if all earlier tokens (if any) had produced a match result. The token cell might also include delaying means to synchronise signal and result throughput. In addition, the token cell must incorporate interconnection means to provide for linkage to other cells, providing for further token comparison or for operations to be performed. The interconnection arrangements determine the ease with which the token cell can be combined with other token or operator cells and the range of regular expressions which can be matched.

It is an object of the present invention to provide regular expression recognition apparatus having an alternative form of interconnection arrangement.

The present invention provides an apparatus for recognition of regular expressions, the apparatus including a token cell comprising:
(1) means for comparing an input character signal with a stored reference signal,
(2) a result input,
(3) result output generation means arranged to indicate a match if an input result match indication has been received and character and reference signals have matched, and
(4) through connections arranged to provide interconnections between other cells.

The invention provides the advantage that the token cell is adapted (as will be described) for interconnection to other token and operator cells in a line which reproduces the order of terms and operators in a corresponding regular expression to be matched. Accordingly, regular expressions may be mapped into matching circuits with great ease. Moreover, there is no requirement for subsequent design of circuit interconnections as in the prior art.

One bit input character and reference signals may conveniently be compared by a NOT Exclusive OR gate, and the reference signal may be stored in a shift register. An AND gate may be employed to generate a result output.

The apparatus of the invention may also include three union operation cells comprising:
(1) a left bracket cell having
  (a) a result input connected to two result outputs,
  (b) a false indication output, and
  (c) through connections for interconnecting other cells;
(2) a union operator cell having
  (a) a result input connected to one input of an OR gate having a second input arranged to receive an earlier union result if available or a left bracket false indication otherwise,
  (b) an OR gate output,
  (c) two result outputs arranged for connection to a result output of a left bracket cell, and
  (d) through connections for interconnecting other cells; and
(3) a right bracket cell having
  (a) a result input connected to an input of an OR gate having a second input arranged for connection to a union operator cell OR gate output,
  (b) a result output connected to the OR-gate output,
  (c) through connections for interconnecting other cells.

As will be described, the union operation cells are designed to be located with respect to token and operation cells in precisely the same order as the corresponding terms in a regular expression irrespective of type. Left and right bracket cells may be arranged in a nested manner in precise correspondence to algebraic expressions incorporating nested brackets. Both the form of the union operation cells and their through or bypass connections provide for nesting without needing ancillary connections as in the prior art.

In a preferred embodiment, the invention also includes two star operation cells comprising:
(1) a left star bracket cell having
  (a) a result input connected to one input of an OR gate,
  (b) a result output and a feedforward line connected to the OR gate output,
  (c) a feedback line connected to a second input of the OR gate,
  (d) through connections for interconnecting other cells; and
(2) a right star bracket cell having
  (a) a result input connected both to one input of an OR gate and to a feedback line for connection to a second input of a left star bracket cell OR gate,
  (b) a result output connected to the OR gate output,
  (c) a signal input connected to a second input of the OR gate and arranged for connection to a result output of a left star bracket cell, and
  (d) through connections for interconnecting other cells.

These cells provide for the star operation and its nesting with union or other star operations.

The invention may include a further star operation cell, a star operator cell for location between star bracket cells, and including:
(a) a first OR gate having a result input and a second input for connection to a result output of a left star bracket cell, the OR gate providing two result outputs for connection to a subsequent cell and to a right star bracket cell OR gate input respectively,
(b) a second OR gate having a first input connected to the first OR gate result input, a second input connectable to a feedback line from a right star bracket cell, and an output connectable to a left star bracket cell feedback line, and
(c) through connections for interconnecting other cells.

The star operator cell provides for a simplified form of expression for multiple star operations, as will be described.

Operation cells of the invention preferably include through connections for routeing character signals between token cells.

In order that the invention might be more fully understood, embodiments thereof will now be

described, by way of example only, with reference to the accompanying schematic functional drawings, in which:

Figure 1 is a circuit diagram of a token cell of the invention,

Figure 2 provides circuit diagrams of union operation cells of the invention,

Figure 3 illustrates an arrangement of the cells shown in Figs. 1 and 2,

Figure 4 is a circuit diagram of star operation and token cells of the invention,

Figure 5 is a circuit diagram of a star operator cell of the invention, and

Figure 6 illustrates use of the star operation cells of Figs. 4 and 5.

Referring to Fig. 1, there is shown a one-bit token cell 10 comprising a character signal input 11 connected to an output 12 and to an input 13 of a NOT Exclusive OR gate 14. A shift register memory 15 storing a one-bit reference character is connected to a second input 16 of the NOT EXOR gate 14. A result input 17 is connected to a one clock cycle delay unit 18 and thence to an input 19 of an AND gate 20 having a result output 21. The NOT EXOR gate 14 has an output 22 connected to a second input 23 of the AND gate 20. The cell 10 also includes three upper and nine lower through connections such as 24 and 25 respectively for interconnecting operation cells, as will be described later. The cell 10 may have more or fewer through connections than illustrated, in accordance with the degree of maximum complexity of regular expression to be matched. Operation of the delay unit 18 is controlled by a system clock (not shown).

The function of the token cell 10 is as follows. On successive clock cycles it receives successive one bit character signals at input 11 which are passed to subsequent token and operation cells (not shown) via output 12. The character signal is compared with the stored one bit reference character by the NOT EXOR gate 14, which gives a 1 output from input of two zero bits or two 1 bits and a 0 output otherwise. The 1 output indicates a true or match result, and the 0 output a false or unmatched result. A one bit result signal obtained one cycle earlier from a preceding token or operation cell (not shown) is delayed by one clock cycle in the delay unit 18. On the subsequent clock cycle, the AND gate will receive the output of the delay unit 18 and the NOT EXOR gate 14. If both these outputs are 1, indicating matches achieved both by any earlier token cells and by the illustrated token cell 10, then the result output 21 of the AND gate 20 will be 1 indicating a match. If either of the signals on result input 17 or the NOT EXOR gate output 22 were to be 0, indicating a non-match for either, the result output 21 would provide a 0 indicating a non-match.

To test for matching of a concatenation of a series of one-bit characters, such as for example an eight-bit word, eight token cells of the form 10 may be connected in series. Since each result input is connected to the result output of the preceding cell, each cell indicates a match only when all preceding cells (if any) have matched. If the first bit matches on a first clock cycle, the second cell will receive a 1 result input allowing it to test the second bit on the second cycle and so on. After eight clock cycles the result would emerge at the result output 21 of the eighth cell. It should be noted that the result input 17 of the first cell must be initialised with a 1 or match indication before input of the first character.

Referring now to Fig. 2, there are shown three cells 30, 31 and 32 for carrying out the union operation. The cells 30 to 32 respectively represent a left bracket (, a union operator cell, ie logical OR or +, and a right bracket). Each of the cells 30 to 32 is associated with a parameter N indicating the degree of bracket nesting introduced by each pair of left and right bracket cells 30 and 32. In a regular expression, the outermost union bracket pair would have N = 1, the next outermost pair N = 2 and so on. For the cells 30 to 32 illustrated, N = 3.

The left bracket cell 30 has character and result inputs 33 and 34 connected directly to character and result outputs 35 and 36. The cell 30 introduces an Nth OR-path line 37 connected to the result input 34. (N − 1)th and (N − 2)th OR-path bypass lines 39 and 40 are provided. The cell 30 also contains (N − 1)th and (N − 2)th OR-result bypass lines 41 and 42, and introduces an Nth OR-result line 43 connected at 44 to a false (F) or non-match indication given by logic 0.

The union operator cell 31 has a character input 50 connected to a character output 51 and a result input 52 connected to one input 53 of an AND gate 54. An Nth OR-result input line 55 is connected to the second input 56 of the AND gate 54, which provides an Nth OR-result output line 57. An Nth OR-path throughput line 58 is also connected to a result output 59. (N − 1) and (N − 2)th OR-path and OR-result bypass lines 60 to 63 are provided.

The right bracket cell 32 has a character input 70 connected to a character output 71, and a result input 72 connected to an input 73 of an AND gate 74. An Nth OR-result line 75 is connected to the second input 76 of the AND gate 74, which also furnishes a result output at 77. An Nth OR-path line 78 is terminated without connection at 79. (N − 1)th and (N − 2)th OR-path and OR-result bypass lines 80 to 83 are provided.

The cells 30 to 32 are connected together via intervening cells as indicated by chain lines. The cells 30 to 32 may also include additional bypass lines (not shown) for star operations, to be described later.

It is important to note that the union operation can only be carried out in accordance with the invention by means of left and right bracket cells 30 and 32 in addition to the operator cell 31. Unions must therefore be represented as eg (A + B), not merely A + B as may be the case in mathematical representations.

Referring now also to Fig. 3, there is shown the regular expression (A + B(C + D + E)) represented in circuit form and indicated generally by 90. The circuit 90 illustrates the use and operation of the union operation cells 30 to 32 of Fig. 2. It incorporates a left bracket cell 91, a character A token cell 92, a union operator cell 93, a character B token cell 94 and a second left bracket cell 95. These are followed by a character C token cell 96, a second union operator cell 97, a character D token cell 98, a third union operator cell 99, a character E token cell 100 and two right bracket cells 101 and 102. For clarity and convenience of illustration, only those OR-path and OR-result bypass lines (cf Figs. 1 and 2) which are in use are shown, and they are rearranged in position somewhat. This has no effect on circuit operation. All the cells 91 to 102 are controlled by the same system clock (not shown).

There are two pairs of nested brackets in the expression (A + B(C + D + E)), these being represented by bracket cell pairs 91–102 and 95–101. The nesting degree parameter N is accordingly equal to 2 within bracket cells 95–101 and 1 otherwise. The expression represents a match to be obtained either with A, or with B when concatenated with C or D or E.

The arrangement of Fig. 3 operates as follows. It will be assumed that the result input 110 of the left bracket cell 91 receives a permanent 1 input signal during character input.

Consider a sequence of two characters input to the left bracket cell character input 111, one character per clock cycle. By virtue of the through character signal line connection 112 to the right bracket cell character output 113, each character in turn is input to all cells 91 to 102 simultaneously. This is referred to in the prior art as broadcasting the characters. The initialisation match signal passes from the cell 91 to the delay unit 114 of token cell 92, and thence one cycle later to the corresponding AND gate 115 whilst the initial input character is present on character signal line 112. The output at 116 of the AND gate 115 will be 1 or 0 according to whether or not the character A stored in token cell 92 matches the initial input character. The output signal at 116 is fed as an input to the OR gate 117 of union operator cell 93, the OR gate 117 being furnished with a second and permanently false or 0 input from left bracket cell 91 via an OR-result line 118. An OR gate 119 in the second right bracket cell 102 receives as an input the output signal from the union operator OR gate 117 via an OR-result bypass line 120. Since one of its inputs permanently receives a false indication or logic 0, the union operator OR gate 117 provides a 1 or 0 output according to whether AND gate 115 has a 1 or 0 output at 116, ie whether or not character A has matched. Accordingly, if character A matches, the second right bracket cell OR gate 119 will receive a 1 input on OR-result line 120 and deliver a 1 or match output at 121 irrespective of the signal at its other input 122. This fulfils the first requirement of the regular expression to be matched, that a match should be indicated if A matches the first character input in a sequence of two characters. Similarly, the output at 121 will indicate a match if A matches the second of the two characters.

If character A is not matched by the first or second input character, cells 94 to 101 assume significance. The initial result input of true or logic 1 at 110 is fed via an OR-path bypass line 123 and a connection 124 to the delay unit 125 of character B token cell 94. The character signal on line 112 is compared by the cell 94 with character B. The AND gate 126 of cell 94 will provide a true or false output according to whether or not a match is achieved with character B, since its two input signals are 1 from delay unit 125 and 1 or 0 for match or non-match. Comparison with character B by cell 94 occurs on the same cycle as comparison with character A by cell 92, since the initialisation result input (true) from 110 experiences a one clock cycle delay in both cases by virtue of units 114 and 125. The result output of AND gate 126 is fed via a line 128 in left bracket cell 95 to the delay unit 129 of character C token cell 96. This result output is also fed via OR-path bypass line 130 to the delay units 131 and 132 of characters D and E token cells 98 and 100. The token cells 96, 98 and 100 accordingly operate simultaneously and one clock cycle later than character A and B token cells 92 and 94.

If the first character of the expression matches character B, AND gate 94 provides a 1 or match result output to become the result input to token cells 96, 98 and 100. One clock cycle later, any of the AND gates 133, 134 and 135 of the cells 96, 98 and 100 respectively will produce a match or 1 output of its stored character matches the second character of the expression. A 1 output from AND gate 133 (character C match) produces successive 1 outputs at OR gates 136, 137, 138 and 119 in cells 97, 99, 101 and 102 respectively. A 1 output from AND gate 134 (character D match) produces successive 1 outputs from OR gates 137, 138 and 119. Finally, a 1 output from AND gate 135 (character E match) produces successive 1 outputs from OR gates 138 and 119. As has been said, a 1 output from OR gate 119 indicates an overall recognition of the regular expression. Accordingly, if character A does not produce a match with either of the first two characters, the circuit 90 will still indicate a match if the characters match BC, BD or BE. The circuit 90 therefore fulfils the requirement to match the

regular expression (A + B(C + D + E)).

The use of shift registers for storage provides a particularly convenient means of loading and reprogramming reference characters in a regular expression. The registers may be connected together in series and loaded or reprogrammed simply by passing the appropriate bit-serial string along them.

There are a number of points to note regarding the circuits of Figs. 1, 2 and 3. They have been described with reference to one bit characters, but multiple bits in parallel may be accommodated merely by employing buses for character signal lines and providing for example multi-bit character stores and comparing means. Alternatively, as has been said, the circuits are operative on bit-serial data by concatenation, ie connecting token cells in series. It will also be observed that right bracket cell 32 incorporates a redundant Nth OR path line 78 left unconnected at 79, and connected to line 58 in cell 31. Nth OR-path line 58 is fed out of cell 31 to provide for a possible subsequent union operator within the same pair of brackets, cf union operator cell 99 following union operator cell 97 in Fig. 3. However, an Nth OR-path line need not in general be connected through to a right bracket cell. It has been described thus simply to enhance the symmetry of the circuits in Fig. 2 as an aid to clarity.

It will be appreciated that the regult expression (A + B(C + D + E)) contains redundancy if A to E are all one-bit characters. If so, two of the characters C, E and E must be the same binary digit, since each character can only be 0 or 1. One of these characters could accordingly be omitted. However, the redundancy vanishes if each of C to E at least represents a two or more bit serial character, ie C_1C_2 to E_1E_2 where C_1 etc are one bit characters. Moreover, the expression was chosen to exemplify inter alia two or more union operators within a bracket pair.

It should be noted that there are situations in which spurious matches can be obtained. Consider the case of the regular expression of serial three-bit characters ABC, where A = 101, B = 011, and C = 100. An input string 101011100 will accordingly match. However, an input string of four three-bit characters such as 001010111001 will also match spuriously, a combination of the two inner characters with parts of the outer characters providing the required string. This can be obviated by making the result input of the character A token cell true once every three cycles. Each true result input is synchronised to occur one cycle before input of a respective leading bit of a character to allow for the effect of delay units such as 18 in Fig. 1. This procedure ensures that only whole characters can be matched. However, in comparatively rare cases this may not be enough to avoid spurious matches. In particular, in searching a database for the word CARBON, a spurious match would be obtained with the first six letters of CARBONATE. This is easily avoided in practice by enclosing each word sought in space characters defining the beginning and ending of the word. Token cells incorporating the appropriate digital space character would be employed in the matching circuit. Other distinguishing characters could also be employed.

Referring now to Fig. 4, there are shown two cells 150 and 151 for implementing the star operation, the cells being arranged either side of a token cell 152 (character A). The star operation on character A, A*, is represented algebraically as [A] for the purposes of mapping in circuitry. The star operation cells 150 and 151 accordingly represent a pair of left and right square brackets respectively. The cell pair incorporates two upper through connections or feedforward lines 153 and two lower through connections or feedback lines 154. Connections to other cells (not shown) are indicated by chain lines, and a character signal through connection line 155 is provided. Left bracket cell 150 contains an OR gate 156 having a result input 157 and a second input connected to a third feedback line 158. The OR gate has a result output line 159 connected to a third feedforward line 160. Right bracket cell 151 has an OR gate 161 receiving inputs from a result line 162 and the third feedforward line 160. An OR gate output line 164 provides a result output. Result line 162 is also connected to the third feedback line 158. The square bracket cell pair 150 and 151 are associated with a parameter M analogous to the parameter N of cells 30 to 32 in Fig. 2. For cells 150 and 151 M = 3, ie they respectively introduce and remove the third feedforward line 160, and remove and introduce the third feedback line 158.

The cells 150 and 152 operate as follows. Cell 150 may receive a 1 result input at 157, which may be an initialisation pulse or a match result from preceding cells (not shown). In consequence, the output line 159 and feedforward line 160 will carry a 1 input to OR gate 161, the output of which will also be 1 indicating a match. This occurs virtually in synchronism with the initial result input at 157. If alternatively cell 150 receives a 0 input on a second clock cycle following a 1 input on the preceding or first clock cycle, token cell 152 assumes significance. Delay unit 165 in token cell 152 will provide a 1 input to AND gate 166, since the delay ensures that the AND gate 166 receives the first clock cycle result input to cell 150 when the second clock cycle character input is present on line 155. If this second character matches character A stored in cell 152, AND gate 166 will provide a 1 output. A 1 signal consequently appears at the output 164 of the OR gate 161 irrespective of its other input on feedforward line 160. The result input signal on result line 162 also passes via third feedback line 158 to OR

gate 156, so that its output at 159 and the input to delay unit 164 become 1 also. This allows token cell 152 to attempt to match the character input to line 155 on the third clock cycle. A match on the third clock cycle allows an attempt to match on the fourth clock cycle and so on for any number of cycles. Consider the regular expression ZA* implemented as Z[A]. An initial character Z would be matched by a token cell (not shown) preceding cell 150. If, on the following cycle, the input signal were to have the value A, it would not be matched by the character Z token cell and character A token cell 152 would come into play. Cell 152 would then match character A and any number of successive As. The cells 150 and 151 accordingly produce the regular expression star operation when enclosing a token cell therebetween. By a similar analysis to that given with reference to Figs. 1 to 3, it will be evident that cells of the form 150 and 151 also provide the star function when employed in combination with union operation cells, concatenation of token cells and nesting of star operation cells. The feedback and feedforward lines 153 and 154 provide for nesting within up to two other pairs of square bracket cells; eg the cells 150 and 151 may be employed to carry out the star operations for the regular expression:

[((A + B)[(C + [DEF] + G)] + H)]

However, in addition to lines 153 and 154, further pairs of through connections (not shown) bypassing union operation and token cells would be required to provide OR path and OR result lines for the union operations, as described with reference to Figs. 2 and 3.

Referring now to Fig. 5, there is shown a star operator cell 170 for simplifying mapping of multiple star operations. The cell 170 has a character signal throughput line 171 and a result input 172 connected to one input 173 of an OR gate 174. A star operation feedforward input line 175 is connected to the other input 176 of the OR gate 174. A star operation feedforward output line 177 is connected to the OR gate output 178, which also provides a result output. The result input 172 is also connected to one input 179 of an OR gate 180. The OR gate 180 has a second input 181 connected to a star operation feedback line 182 (cf line 158 in Fig. 4), and an output connected to a feedback line 183. The cell 170 also contains two additional star operation feedback lines 184 and two additional feedforward lines 185.

The cell 170 is employed as follows. The regular expression (A*B*)*, representing an arbitrary number of Bs following an aribitrary number of As and the combination repeated an arbitrary number of times, could be mapped in circuitry in accordance with the invention as [[A][B]]. Alternatively, the regular expression (A*B*)* may be implemented as [A*B], where the square brackets correspond to cells 150 and 151 and the central star operator corresponds to cell 170. This results in an arrangement similar to the union operation cells 30 to 32.

Referring now also to Fig. 6, there is shown the regular expression [A*B] mapped as a circuit indicated generally by 190 in accordance with the invention. The circuit 190 contains a series arrangement of left square bracket cell 191, character A token cell 192, star operator cell 193, character B token cell 194 and right square bracket cell 195. The circuit 190 would be equivalent to that shown in Fig. 4 if star operator cell 193 were to be removed, the concatenation AB replacing A in the earlier example. The effect of cell 193 is to introduce an additional OR gate 196 into a feedforward line having parts 197 and 198, and which would otherwise have connected cells 191 and 195. In addition, OR gate 196 receives the result output line 199 of token cell 192, which would otherwise have been connected to the result input line 200 of token cell 194. The OR gate 196 receives as inputs the result outputs of left square bracket cell 191 and character A token cell 192, and generates a result output to become a result input to character B token cell 194 and a feedforward input to OR gate 201 of right bracket cell 195. Furthermore, the cell 193 also introduces an OR gate 202 into a feedback line having parts 203 and 204, which line would otherwise have directly connected cells 195 and 191. The OR gate 202 receives as inputs the results inputs of cells 193 and 195 via lines 205 and 203, and its output on line 204 becomes the input to OR gate 207 in cell 191. The result output 208 of cell 195 will be 1 whenever the result input 109 of cell 191 is 1, since OR gates 207, 196 and 201 must in this case produce successive 1 outputs. If a 1 result input to cell 191 on a first clock cycle is succeeded by a 0 result input on the next or second clock cycle, then A and B token cells 192 and 194 assume significance. Either of cells 192 or 194 will produce a 1 output if its respective stored character matches that on signal throughput line 210 during the second cycle. OR gate 201 (character B match) or both OR gates 196 and 201 (character A match) will produce a consequent 1 output. In addition, a 1 result output from either of token cells 192 and 194 will produce a 1 output from OR gate 202 to be fed via line 204 to OR gate 207. OR gate 207 accordingly furnishes token cells 192 and 194 with a 1 result input to allow each to test for additional matches. This continues for an arbitrary number of successive cycles provided that each cycle produces a match to A or B or an initial input result at 209 indicates a match (corresponding to no As or Bs). Accordingly, the circuit 190 indicates a match for a combination of an arbitrary number of As followed by an

arbitrary number of Bs repeated an arbitrary number of times, as originally required to match (A*B*)* or its circuit mapping [A*B]. This is achieved using the star operator cell 193 (or 170) without nested and concatenated starring arrangements, ie [[A][B]].

Referring now to Fig. 1 once more, as has been said, the token cell 10 incorporates three upper and nine lower through connections such as 24 and 25 respectively. Comparison of Fig. 1 with Figs. 2 to 6 illustrates their use. The through connections are provided for interconnection of union and/or star operation cells. This allows a maximum of (12−2p) feedforward and feedback lines for the star operation and 2p OR path and OR result lines for the union operation, where p is equal to 0, 1, 2 . . . 6. However, referring now to Fig. 3 once more, it will be noted that token cells 92 and 94 require only two bypass or through connections 123 and 118/120, whereas token cells 96, 98 and 100 require four through connections 120, 123, 130 and the lines 139 linking OR gates 136, 137 and 138. Accordingly, the minimum number of through connections any cell requires depends both on the degree of complexity of the relevant regular expression (maximum of 2(N + M), sum of union and star parameters), and on the position or degree of nesting of the cell in the mapped expression. If the invention were to be implemented as an integrated circuit, redundant through connections might be omitted. However, if discrete components for manual assembly were to be required, it may be convenient to provide each cell with a sufficient number of through connections to ensure its operation in any position. This also reduces the number of types of cell required, since for example it would not be necessary to provide token cells with differing numbers of through connections.

In general, to represent arbitrary regular expressions by means of the invention, it is necessary for cells to be parameterised to implement any connections needed. Each cell requires two parameters, N and M, N referring to the depth of nesting of a corresponding union operation, and M to the depth of nesting of a corresponding star operation. In addition, a language is required in which the recognition apparatus of the invention can be defined. The language might be used as the input to a hardware synthesis or integrated circuit design program, which constructs the necessary cells. The language and circuitry of the invention it defines would have a one-to-one mapping onto the generally accepted notation for regular expressions. For the purposes of this specification, the language is defined with reference to the following set of rules:

(a) Each set of unions must be enclosed within round brackets.

(b) Each starred sub-expression must be enclosed within square brackets, except in the case described with reference to Figs. 5 and 6, in which the star is included within square brackets.

(c) Parameters N and M must be associated with each token according to the relevant union and star depths.

Consider the regular expression: ((A + B)* + C + (D*E*)*) (1)

When partially transformed this becomes: ([(A + B)] + C + [D*E])

The parameters for each cell are evaluated as follows:

	([(A + B)] + C + [D*E])
union parameter level N:	1 1 2 2 2 2 2 1 1 1 1 1 1 1 1 1
star parameter level M:	0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0

The language may be used to formulate a hardware synthesis program, which would construct the required circuitry by concatenating cells in the order given by the language. The construction of each cell is governed by the value of its parameters. Subsidiary interconnection of cells as required by Foster and Kung is not required.

It should be noted that a hardware description language could be used to describe the recognition circuit once the expression has been transformed. However, such a language would have to support the concept of delivery of functions from other functions, so that signals might pass in both directions between square bracket cells.

In many applications it is desirable to implement the invention in as direct a manner as possible, and to map it in circuitry as closely as possible. However, in some cases it may be desirable to attempt to minimise a regular expression in terms of the hardware required to implement it. In particular it can be shown that any starred bracket of unions can be implemented in one level of starring if desired. This arises from use of the identity (A + B + C . . . + Z)*=(A*B*C* . . . Z*)*, and the fact (A*B*C* . . . Z*)* may be implemented in one level of starring as described earlier with reference to Figs. 5 and 6.

A result of this manipulation is that any regular expression can be reduced to the form: $(A_1 + B_1 + \ldots + Z_1)(A_2 + B_2 + \ldots + Z_2) \ldots (A_n + B_n + \ldots + Z_n)$, where A_i, B_i etc, are regular expressions not involving the union operation. The net effect of this is that the depth of nesting of union operators is limited to 1, and so it is not absolutely essential to provide a parameter for the current depth of union nesting. This reduction process, however, may not result in an efficient hardware implementation. Consider the expression: (ABCDE(F + G))*, this may be written: (ABCDEF + ABCDEG)*, or alternatively: [ABCDE-

F*ABCDEG), which uses many more token cells than necessary. A combination of manipulation together with the use of parameters for both unions and starring is accordingly suggested.

The invention has the following advantages:

1. The expression maps onto the cell types in terms of symbols and positions of symbols;
2. Once the parameters N and M have been evaluated, the required cells can be placed adjacent to each other. Connections between cells are made automatically with no further connections being required;
3. The approach lends itself to programmability for arbitrary expressions;
4. Cell combinations may easily be cascaded in series or parallel in order to allow more complex expressions to be implemented;
5. Circuit design could be automated;
6. The throughput rate is likely to be twice that of Foster and Kung's apparatus, which requires the insertion of a 'zero token' between each signal token to synchronise signal values with corresponding match results.

CLAIMS

1. An apparatus for recognition of regular expressions, the apparatus including a token cell comprising:
(1) means for comparing an input character signal with a stored reference signal,
(2) a result input,
(3) result output generation means arranged to indicate a match if an input result match indication has been received and character and reference signals have matched, and
(4) through connections arranged for interconnecting other cells.

2. An apparatus according to Claim 1 and including three union operation cells comprising:
(1) a left bracket cell having
(a) a result input connected to two result outputs, and
(b) through connections for interconnecting other cells;
(2) a union operator cell having
(a) a result input connected to one input of an OR gate having a second input arranged to receive an earlier union result if available or a false indication otherwise,
(b) an OR gate output,
(c) two result outputs arranged for connection to a result output of a left bracket cell, and
(d) through connections for interconnecting other cells; and
(3) a right bracket cell having
(a) a result input connected to an input of an OR gate having a second input arranged for connection to a union operator cell OR gate output,
(b) a result output connected to the OR-gate output,
(c) through connections for interconnecting other cells.

3. An apparatus according to Claim 1 or 2 including two star operation cells comprising:
(1) a left star bracket cell having
(a) a result input connected to one input of an OR gate,
(b) a result output and a feedforward line connected to the OR gate output,
(c) a feedback line connected to a second input of the OR gate,
(d) through connections for interconnecting other cells; and
(2) a right star bracket cell having
(a) a result input connected both to one input of an OR gate and to a feedback line for connection to a second input of a left star bracket cell OR gate,
(b) a result output connected to the OR gate output,
(c) a signal input connected to a second input of the OR gate and arranged for connection to a result output of a left star bracket cell, and
(d) through connections for interconnecting other cells.

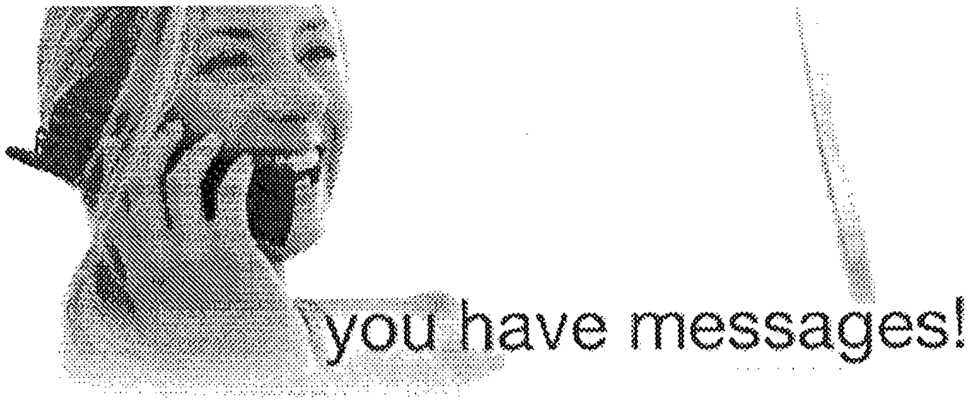
4. An apparatus according to Claim 3 including a star operator cell comprising:
(a) a first OR gate having a result input and a second input for connection to a result output of a left star bracket cell, the OR gate providing two result outputs for connection to a subsequent cell and to a right star bracket cell OR gate input respectively,
(b) a second OR gate having a first input connected to the first OR gate result input, a second input connectable to a feedback line from a right star bracket cell, and an output connectable to a left star bracket cell feedback line, and
(c) through connections for interconnecting other cells.

5. A regular expression recognition apparatus substantially as herein described with reference to the accompanying drawings.

Printed in the United Kingdom for Her Majesty's Stationery Office, Dd 8818935, 1985, 4235.
Published at The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

Dildine, Stephen

From: Personals Plus [p2p_notification@phx.com]
Sent: Friday, May 21, 2004 11:33 AM
To: Dildine, Stephen
Subject: Wilma, 58, left you a MESSAGE



washingtonpost.com
Personals

Dear cavevisit,

Hi! This is Victoria again and I have GREAT news! You have a **voice mail message** waiting for you. Someone has seen your profile and taken the time to leave a message for you because they want to meet you. You joined to meet new people, and now it's working!

When you signed up for our service, you automatically received a Membership ID number and password for the voice component of Personals Plus that allows you to meet even more people!

Just follow the instructions below to meet the gal who left you the message, it's EASY!

To listen to your new message, just call toll free, 1 (866) 803-0637

When you call, follow the instructions as an existing member by pressing 2 at the main menu and logging in with the following:

- Your Membership ID = 892167
- Your Password = 4264

When you call, you will be able to listen to your messages by using a major credit card for $1.99 per minute. You can save up to 30% by prepaying for blocks of time.

Or, you can call 1 (900) 226-8928, also for $1.99 per minute, to have the call charged directly to your phone.

Thanks!

Victoria

You must be 18 or older to use this service.

To be removed from our service, click here, or cut and paste the following text into your browser.
http://www.tpiconnect.com/celldate/members/confirm.asp?MemberID=892167